

Испит из Релационих база података, септембар 1 2020. год. (И смер)

Број индекса	Име и презиме

Задаци се раде 240 минута. Максималан број поена је 70. Број поена на испиту се израчунава тако што се саберу освојени поени по задацима, збир подели са два и заокружи. Број поена по задацима је:

Задатак	1	2	3	Збир 1-3	4	5	6	7	Збир 4-6	Укупно	
Максимално	14	16	20	50	Збир/2	15	15	30	30	90	Збир/2
Освојено											

- Написати SQL упит којим се издвајају информације о свим студентима смерова *Informatika* и *Racunarstvo i informatika* који су са оценом 10 положили испит који носи 9 ЕСПБ поена у испитном року у коме нису пали нити један испит. Издвојити индекс, име, презиме као и датум уписа студента на факултет. Резултат упита приказати у опадајућем редоследу у односу на датуме уписа.
- Написати SQL упит којим се за сваког студента који је (1) рођен у Краљеву или Чачку, (2) уписао као самофинансирајући студент школску 2010/11. годину и (3) излазио на испите у највише 10 различитих испитних рокова, приказују извештај чији су називи колоне и садржај:

Назив колоне	Садржај
<i>Redni broj</i>	4 цифре индекса мање тежине
<i>Име</i>	Име и презиме студента
<i>Иницијали</i>	иницијале у формату: <i>прво слово имена, тачка, прво слово презимена, тачка</i>
<i>Број рокова</i>	број различитих рокова у којима је тај студент излазио на испите
<i>Бодови</i>	укупан број бодова које је скупио успешним полагањем испита
<i>Последњи положен</i>	датум последњег положеног испита

За статус студента у школској 2010/2011. години узети статус евидентиран оног датума када је студент уписао ту школску годину. Уколико недостају вредности за податке нумеричког типа приказати 0, у осталим случајевима приказати NULL.

- (а) Написати SQL наредбу којом се прави табела *studenti_generacije* која садржи податке о најбољим студентима за сваки смер и сваку школску годину у којој је било дипломираних студената са колонама:
 - indeks* - индекс студента;
 - id_smera* - идентификатор смера који је студент студирао;
 - sk_godina* - школска година у којој је студент дипломирао у облику *календарска година на почетку школске године / календарска година на крају школске године* (нпр. 20018/2019);
 - prosek* - просечна оцена студента из положених испита;
 - koeficijent* - коефицијент успешности студирања студента који се рачуна по формули $\frac{\text{просек} * 100}{\text{укупан број дана студирања}}$. Укупан број дана студирања је број дана протекао од датума уписа на факултет до датума када је студент стекао статус *diplomirao*.
 - najbolji_u_generaciji* - идентификатор да ли је најуспешнији студент који је дипломирао у школској години (без обзира на смер);
 - najbolji_ikad* - идентификатор да ли је најуспешнији студент у свим генерацијама на факултету.
- (б) Написати SQL наредбу којом се у табели *studenti_generacije* постављају примарни кључ и ограничење да вредност колоне *najbolji_u_generaciji* и *najbolji_ikad* може бити само 0 или 1.
- (с) Написати SQL наредбу којом се у табелу *studenti_generacije* уносе подаци о најуспешнијим студентима за сваку школску годину и сваки смер. Студент је најуспешнији на свом смеру у школској години у којој је дипломирао ако има највећи коефицијент успешности међу свим студентима који су студирали исти смер и који су дипломирали исте школске године. Унети индекс студента, идентификатор смера који је студирао, школску годину у којој је дипломирао, просечну оцену и коефицијент успешности.

- (d) Написати SQL наредбу којом се у табели *studenti_generacije*
- за најуспешнијег студента према коефицијенту успешности (или студенте ако их има више) у школској години у којој су дипломирали (без обзира који су смер завршили) поставља вредност у колони *najbolji_u_generaciji* на 1, а за остале студенте на 0.
 - за најуспешнијег студента према коефицијенту успешности (или студенте ако их има више) који је дипломирао поставља вредност у колони *najbolji_ikad* на 1, а за остале студенте на 0.
- (e) Написати SQL наредбу којом се из табеле *studenti_generacije* бришу подаци о студентима који нису најуспешнији у школској години у којој су дипломирали.
- (f) Написати SQL наредбу за брисање табеле *studenti_generacije*.

4. а) Детаљно описати ANSI/SPARC архитектуру базе података.
 б) Шта је релациона алгебра?
 в) Наведите интуитивну дефиницију операције потпуног спољашњег спајања, као и случајеве у којима има оправдања за њену примену.
5. Написати кориснички дефинисану функцију *bolji_od_proseka* чији је аргумент *predmet* - идентификација предмета, а вредност листа која садржи имена, презимена и просечне оцене студената за које важи да им је просечна оцена (из свих предмета) већа од вредности (просечна оцена - стандардна девијација оцене свих студената који су положили тај предмет) за предмет чија је идентификација задатак као аргумент функције. Подаци за појединачне студенте треба да буду раздвојени тачком зарезом и сортирани по просеку студента у опадајућем редоследу. Претпоставити да за унету идентификацију предмета постоје студенти који положили тај предмет.
6. Нека је дата релациона променљива $R = \{A, B, C, D, E, F\}$ и скуп F ФЗ:

- | | | | |
|-----------------------|------------------------|-----------------------|------------------------|
| 1) $AB \rightarrow C$ | 3) $BC \rightarrow D$ | 5) $D \rightarrow EF$ | 7) $CF \rightarrow BD$ |
| 2) $C \rightarrow A$ | 4) $ACD \rightarrow B$ | 6) $BE \rightarrow C$ | 8) $CE \rightarrow AF$ |

- а) Одредити нередуцибилни покривач скупа функционалних зависности F .
 б) Одредити све кандидате за кључ релације R .
 в) Испитати да ли је релација R у BCNF и декомпоновати је ако није. У случају да се врши декомпозиција да ли добијено решење доводи до губитка информација и (не)очувања скупа ФЗ? Образложити одговор.

Обавезно образложити СВЕ кораке у раду. Навођење само резултата појединих корака неће бити признато као делимично урађен задатак.

7. а) Напишите наредбе којима се даје дозвола корисницима $K1$ и $K2$ за ажурирање колона *mesto_rodjenja* и *datum_rodjenja* у табели *dosije*, а затим укида дозвола над том табелом за ажурирање колоне *datum_rodjenja* али не и дозвола за ажурирање колоне *mesto_rodjenja* кориснику $K2$.
 б) Која је сврха употребе MQT табела? Која је разлика између MQT табела и погледа?
 в) Нека је дата табела `create table dosije_xml (podatak xml not null)` при чему су изглед и структура слога из табеле

```
<?xml version="1.0" encoding="UTF-8" ?>
  <student broj_indeksa="20140122">
    <ime>Ђорђе</ime>
    <prezime>Ђорђевић</prezime>
    <datum_rodjenja>"28.02.1995"</datum_rodjenja>
    <godina_studija>2</godina_studija>
    <prosek>9.45</prosek>
    <broj_polozenih_ispita>10</godina_studija>
  </student>
```

Написати упит који враћа број индекса, име, презиме, датум рођења, годину студија, број положених испита и просек оцена свих студената чије је име 'Ђорђе', који су на трећој години студија, положили су више од 15 испита и чији је просек оцена већи од 8.5.

Упутство:

Направите нови директоријум на Desktop-у чије је име облика **rbpsep1.brojIndeksa.ime.prezime**. Нпр. студент Марко Марковић са бројем индекса 125/2017 треба да направи директоријум *rbpsep1.2017125.Marko.Markovic*. У том директоријуму оставите решења задатака. Решење сваког задатка оставите у посебној датотеци. Датотеке са решењима назовите редним бројем задатка (нпр. *1.sql, 2.sql ...*). **За сваки задатак оставите тачно једно решење, у супротном задатак неће бити прегледан.**

Задаци 1-3 се решавају над базом података *vstud*, а 4-7 над базом података *mstud*.

Синтаксно неисправна решења за задатке 1-3 носе 0 од предвиђеног броја поена. Грешке у решењима задатака 4-7 које показују непознавање испитне материје повлаче 0 поена на комплетном задатку.

Дозвољена литература се налази у директоријуму `/home/ispit/Desktop/Materijali/rbpKnjige`.